International Journal of Electronics, Communication & Instrumentation Engineering Research and Development (IJECIERD)
ISSN(P): 2249-684X; ISSN(E): 2249-7951
Vol. 5, Issue 3, Jun 2015, 1-10

AN EFFICIENT HYBRID BAT-OPTIMIZED CLUSTERING FOR WIRELESS SENSOR NETWORKS

KOTESWARARAO SEELAM[1], M. SAILAJA[2] & T. MADHU[3]
[1]Department of ECE, JNTUK, Kakinada, Andhra Pradesh, India
[2]Department of ECE, SIET, Narsapur, Andhra Pradesh, India

ABSTRACT

Cluster-based routing protocols offer advantages such as improved power control, reduced control messages, enhanced resource allocation and bandwidth re-usability. Low Energy Adaptive Clustering Hierarchy (LEACH) a cluster-based protocol includes distributed cluster formation. LEACH randomly selects sensor nodes as cluster-heads and rotates them to distribute energy load uniformly among network sensors. LEACH is ambiguous sensor nodes position and network Cluster Head (CH) numbers. This study proposes a hybrid BAT algorithm (HBA) with Differential Evolution (DE) to improve the efficiency and to overcome disadvantages of LEACH Simulation study revealed that the HBA achieved improved throughput, lowered delay and packets retransmission and better data dropped than LEACH.

KEYWORDS: Differential Evolution (DE), Low Energy Adaptive Clustering Hierarchy (LEACH), Throughput

INTRODUCTION

From the beginning of the third Millennium, Wireless Sensor Networks (WSNs) [1] has generated interest in both the industry and in the research field. WSNs are emerging, pervasive computing applications with several small, low power, and intelligent sensor nodes (motes) and one or more base stations. Sensor nodes gather information in varied settings, including battlefields, natural ecosystems, and man-made environments and forward the same to one or more base stations [2]. Although WSNs, enable new applications and thereby new markets, their design is affected by several constraints that need new paradigms. In fact, sensing, processing, and communication under restricted energy ignites a cross-layer design approach that require jointed efforts of distributed signal/data processing, medium access control, and communication protocols.

Sensor network routing is challenging owing to the characteristics that distinguish it from the current communication and wireless ad-hoc networks [3]. First, it is impossible to build a global addressing scheme for deploying the vast number of nodes. Hence, classical IP-based protocols are not applicable for sensor networks. Second, contrary to the typical communication networks, most senior network applications require a sensed data flow from multiple regions (sources) to a specific sink. Third, the generated data traffic has redundancy as multiple sensors generate same data within the vicinity of a phenomenon. This redundancy can be exploited by routing protocols to improve the energy and bandwidth use. Fourth, sensor nodes are highly constrained with respect to power transmission, on-board energy, processing capability, and storage; therefore, there is a need for proper resource management. These constraints are widely addressed by flow control and cluster-based routing.

Error control/Flow control [4] in a TCP/IP model data link layer detects errors in the received frames and retransmission requests of frames, whereas flow control determines data to be transmitted in a specific period. All networks do not run at the same speed, and therefore the flow control is necessary to control the data amount sent by a device so that the receiving device can accept and handle the data. The sliding window method and stop-and-wait method suits flow control. Flow control [5] prevents this situation by limiting the sender from sending data beyond the handling capacity of the receiver:

- Basic flow control principle applies well-defined rules to the time of transmitting of the frame by the sender
- These rules prohibit frames from being sent until the receiver grants permission for the same
- Data link layer uses this feedback-based flow control.

Cluster-based routing protocols offers advantages [6] such as improved power control, reduced control messages, enhanced resource allocation, and bandwidth re-usability. However, cluster-heads are fixed in the current cluster-based protocols. Therefore, overall network life decreases in a sensor network environment because cluster-heads involve energy-intensive processes and energy consumption is more than that in non-cluster-head nodes. Several cluster-based sensor networks adapted routing protocols were proposed to incorporate the advantages of traditional cluster-based routing protocols.

LEACH [7] is a popular cluster-based protocol with distributed cluster formation. LEACH selects some sensor nodes randomly as cluster-heads and rotates the same to distribute energy load uniformly among the network sensors. In LEACH, cluster-heads compress the data from nodes that belong to the respective clusters and send an aggregated packet to BS in order to reduce information to be transmitted to BS.

LEACH protocol allows the data transmission phase last for a specific time, followed by entrance into a new Cluster Head (CH) election round. The time length of the round evidently influences the LEACH protocol performance. In order to decrease the set up of the overhead phase, the time length of the round should be increased, which also increases the time of data transmission. However, prolonging the time length of the round also increases the CH energy consumption, which leads to the early death of some nodes, shortening the WSN life. Therefore, for the time length configuration of the round, a tradeoff exists between the life and throughput.

To overcome the disadvantages of LEACH and to improve the efficiency of the cluster formation, this study proposed a hybrid BAT algorithm (HBA) for cluster formation with CH selection. The proposed hybrid algorithm was based on BAT and Differential Evolution (DE) optimization techniques. Sect 2 presents the literature survey; sect 3 explains the methodology of LEACH, BAT algorithm, DE, and HBA. Sect 4 presents the results, and, finally, sect 5 concludes the overall performance of work.

LITERATURE SURVEY

WSN cluster-based routing mechanisms were analyzed by Jiang et al. [8]. CH selection, cluster formation, and data transmission of the cluster-based routing protocols were introduced, and their characteristics and performances were compared. Finally, future research issues were indicated. A scheme to enhance LEACH by deploying multiple base stations was proposed by Reddy et al. [9]. Simulations undertook deployment of multiple base stations, followed by analysis within a specific terrain with respect to the total energy consumed.

A new routing approach based on the Ant Colony Optimization (ACO) algorithm in WSN, in which LEACH protocol was applied, was proposed by Sharma et al. [10] to route the sensor networks data packets in order to maximize the energy efficiency and increase the network life. This approach attempted to reduce the effort applied to send redundant data by sensors that were close to others in a sensor network. Simulation showed that the new approach provided optimized solutions regarding efficient energy use and enhanced network life.

An energy-efficient clustering scheme based on grid optimization using Genetic Algorithm (GA) was proposed by Kumar et al. [11], in which a sensing field was divided into virtual grids, each grid representing a cluster. GA optimized grids to equal node numbers in every grid, generating uniform traffic load on each cluster. This step enhanced the network life. The residual energy of nodes was considered during CH selection to balance the network energy.

Low Energy Intelligent Clustering Protocol (LEICP) was proposed by Li et al. [12]. Fitness function balances the energy consumption in clusters according to the residual energy and nodes positions. In each round, a node named auxiliary cluster-head calculated the CH position by using the Bacterial Foraging Optimization Algorithm (BFOA). After data aggregating, the Dijkstra algorithm computed an optimal path. Simulation demonstrated that LEICP prolonged the sensor network life by approximately 62.28% as compared to LEACH and acquired uniform cluster-heads and network messages.

In the work of Parvin and Vasanthanayaki [13], the nodes were allowed to form clusters without any CH selection. CH role was substituted by the mobile sink nodes for data gathering, data aggregation, and communication operations, which increased the lifetime of the sensor nodes. Gravitational Search Algorithm (GSA) is an optimization algorithm that was used for cluster formation and routing data to achieve higher throughput. Simulations were performed with Network Simulator.

Siew et al. [14] achieved an energy-efficient clustering protocol in which selection of CH considered various critical parameters. The LEACH protocol was used for cluster formation that allowed random rotation of CHs, fuzzy logic in Base Station (BS) for suitable CH selection, and Particle Swarm Optimization (PSO) to select a suitable set of CH.

METHODOLOGY

Low Energy Adaptive Clustering Hierarchy (LEACH)

LEACH [15] is a typical hierarchical routing protocol representation that is self-adaptive and self-organized. LEACH protocol uses “round” as a unit, with each round composed of a cluster set-up stage and a steady-state stage. To reduce unnecessary energy costs, steady-state stage should be longer than set-up stage, whose process is depicted in Figure 1.

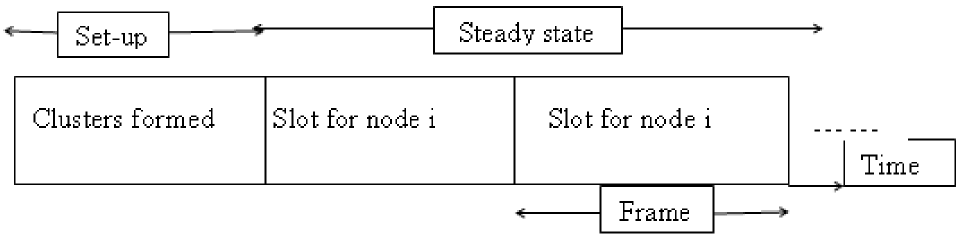

Figure 1: The LEACH Protocol

At the cluster-forming stage, a node randomly picks a number between 0 to 1 and compares this number to the threshold values $t(n)$; if number is less than $t(n)$, it becomes CH in this round, otherwise it becomes a common node. Threshold $t(n)$ is determined by the following equation:

$$t(n) = \begin{cases} \dfrac{p}{1 - p * (r \bmod \frac{1}{p})} & if\ n \in G \\ 0 & if\ n \notin G \end{cases} \tag{1}$$

Where, P is the percentage of network clusters [16] (usually $P = 0.05$), r is the number of election rounds, $r\mathrm{mod}(1/P)$ is the number of nodes elected as CHs in round r, and G is the set of nodes not elected as CHs in round r.

After CH election, CHs broadcast their identity message to non-CH nodes. The latter send a join-REQ message to the nearest CH to join the corresponding cluster. After CH receives all join-REQ information, it produces a TDMA schedule and notifies the cluster member nodes. When a member node receives the schedule, it sends data in own-time slots and remains in the sleep state in the other slots. After data transmission, CH runs a data compression algorithm to process the data and sends the obtained results to the sink node.

An optimal percentage p_{opt} of nodes become CHs in each round. Energy efficiency of some CHs is not optimal because the values may be very similar or located in the edge of the WSN. In this study, LEACH was enhanced by using the hybrid BAT optimization method, wherein BAT algorithm was hybridized using DE for cluster formation and CH selection based on distances and the number of CHs.

BAT Algorithm

A new meta-heuristic search algorithm named BAT algorithm (BA) [17] was developed by Xin-She Yang. Bats are fascinating. The mammals have wings with advanced echolocation capability. Micro bats use a sonar called echolocation to detect prey, avoid obstacles, and locate roosting crevices in the dark. Bats emit a loud sound pulse and listen to the resounding echo from the surrounding objects. Their pulse varies in properties and is correlated with their species-dependent hunting strategies.

If the characteristics of the echolocation of micro bats are idealized, a BAT algorithm is developed. For simplicity, the following approximate rules are used:

- Bats use echolocation to sense the distance and there by deduce the difference between food/prey and background barriers
- Bats fly randomly with a velocity vi at position xi at a frequency f_{min}, varying wavelength λ, and loudness A_0 to search for their prey. They automatically adjust the wavelength (or frequency) of the emitted pulses as well as the pulse emission rate r Є [0, 1], based on the target proximity.
- Although loudness varies in several ways, it can be assumed that loudness varies from a large (positive) A_0 to a minimum constant value Amin. Based on approximation and idealization, the BAT algorithm's basic steps (BA) [18] have been summarized as a pseudo code in the next section.

Pseudo Code of BA

```
Objective function: f(x), x = (x_1,....x_d)^t
Initialize bat population x_i and velocity v_i i = 1, 2,..n
Define pulse frequency f_i at x_i
Initialize pulse rate r_i and loudness A_i
While (t < maximum number of iterations)
Generate new solutions by adjusting frequency and
Updating velocities and location/solutions.
F (rand > r_i)
Select a solution among the best solutions
Generate a local solution around the selected best solution
End if
If (rand < A_i and f(x_i) < f(x*))
Accept new solutions
Increase r_i reduce A_i
End if
Ranks the bats and find current best x*
End while
Display results.
```

There are several reasons for the success of BAT-based algorithm [19]. By analyzing the key features and updating the equations, the following three points/features are summarized:

- **Frequency Tuning:** BA uses echolocation and frequency tuning to solve problems. Although echolocation does not directly mimic the real function, it uses frequency variations. This ability provides some functionality that are similar to the key feature in PSO and harmony search. Therefore, BA possesses advantages over other swarm-intelligence algorithms.
- **Automatic Zooming**: BA offers a major advantage over meta-heuristic algorithms. BA can automatically zoom into a region where promising solutions occur. Zooming is accompanied by automatic switching from explorative moves to local intensive exploitation, leading to quick convergence rate at iterations early stages as compared to other algorithms.
- **Parameter Control:** Several meta-heuristic algorithms fix the parameters by using pre-tuned algorithm-dependent parameters. In contrast, BA uses parameter control, which varies the parameters (A and r) values as iterations proceed, showing a way to automatically switch from exploration to exploitation under optimal solution.

Differential Evolution (DE)

DE [20] belongs to GAs that uses biology-inspired crossover, mutation, and selection operations on a population in order to minimize the objective function over successive generations. As with the other evolutionary algorithms, DE solves optimization problems by evolving a candidate solutions' population by using alteration and selection operators. DE uses floating-point instead of bit-string encoding of population members and arithmetic operations and not logical operations in mutation in contrast to classic GAs.

Let NP denote the parameter vectors (members) x € R^d in a population, where d denotes dimension. To create the initial generation, NP guesses the optimal value of parameter vector using random values between the upper and lower bounds (defined by user) or by using values provided by the user. Each generation involves a new population creation from the current population members {x_i | i =1,…..,NP}, where i is the index vector that constitutes the population. This is accomplished by using differential mutation of the population members. An initial mutant parameter vector $\mathbf{v}_i$ is created through selecting 3 members of population, x_{i1},, x_{i2}, and x_{i3}, randomly. Then v_i is generated as follows:

$$v_i = x_{i_1} + F.(x_{i_2} - x_{i_3}) \tag{2}$$

Where, F is a positive-scale factor, effective values for which are less than one. After the first mutation operation, it is continued until d mutations are made, with a crossover probability of $CR \in [0,1]$. The crossover probability CR controls parameter values fraction copied from mutant. Thus, mutation is applied to each member of a population. When an element of trial parameter vector is found to violate the bounds after mutation and crossover, it is reset so that the bounds are respected. Then, the objective function values associated with children are determined. When a trial vector has equal/lower objective function value than earlier vector, it replaces the previous vector in a population, or the earlier vector remains.

Intuitively, the scheme's effect is that the shape of the population distribution in the search space converges regarding the size and direction to the areas with high fitness. The closer the population gets to the global optimum, the more the distribution shrinks and reinforces the generation of smaller difference vectors [21].

Hybrid BAT Algorithm

This study proposed a new BAT algorithm, called as the HBA [22]. The original BAT algorithm was hybridized by using DE strategies. The fitness is based on the total transmission distance and the number of CHs. The fitness is computed by the following equation:

$$f = \alpha * \left(D_{total} - d_i\right) + \left(1 - \alpha\right) * \left(N_{total} - N_{ch}\right) \tag{3}$$

Where, α is a pre-defined weight, D_{total} is the distance of all nodes to the sink, d_i is the sum of distance of nodes to CHs, N_{total} is the number of nodes in the network, and N_{ch} is the number of CH. The fitness of the node increases as the distance decreases and the number of CHs is less. During initialization, the algorithm randomly selects nodes to be CHs in the network. Based on the fitness function, the algorithm searches for appropriate number of CHs and its location.HBA's pseudo-code is illustrated as follows:

Pseudo Code for H-B-A

```
Objective function f(x), x = (x_1; :::; x_d)^T
Initialize the bat population x_i and v_i for i = 1 : : : n
Define pulse frequency Q_i [Q_min; Q_max]
Initialize pulse rates r_i and the loudness A_i
While (t < T_max) // number of iterations
Generate new solutions by adjusting frequency, and
```

Updating velocities and locations/solutions
If$\left(\text{rand}\left(0;\ 1\right)\ >\ r_i\right)$
Modify the solution using"DE/rand/1/bin"
End if
Generate a new solution by flying randomly
If$\left(\text{rand}\left(0;\ 1\right)\ <\ A_i \text{and } f\left(x_i\right)\ <\ f\left(x\right)\right)$
Accept the new solutions
Increase r_i and reduce A_i
End if
Rank the bats and find the current best
End while
Post process results and visualization

Experimental Results and Discussions

LEACH and the proposed hybrid BAT was compared in this study to determine the performance of throughput, delay, retransmission of packets, and data dropped using MATLAB & OPNET Modeler 14.5, Figures 2 - 5 show the results of the same, respectively.

Table 1: Simulation Parameters

S. No	Parameter	Value
1	Radio propagation range of node	100 m
2	Network area	750 *750
3	Initial energy of each node	2.5 J
4	Number of nodes	75 to 275
5	Traffic type	Constant bit rate(CBR)
6	Simulator	OPNET Modeler 14.5 & Matalab
7	Simulation time	50 sec
8	No. of cluster heads	5(default),10

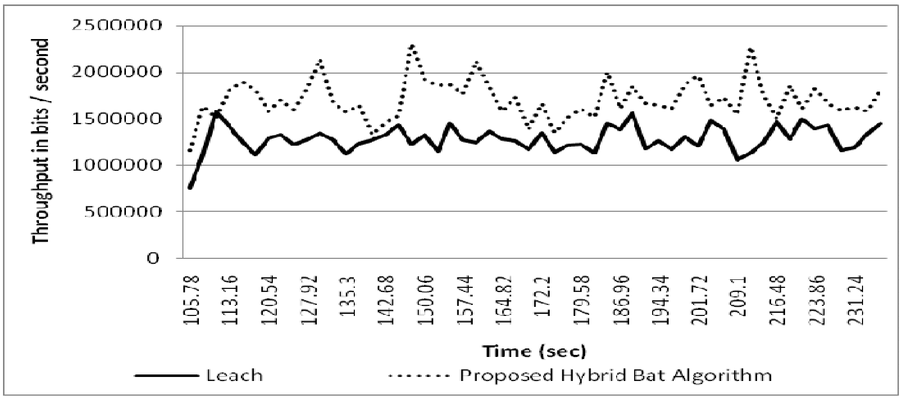

Figure 2: Throughput in Bits/s vs Time(s)

Figure 2 shows that the proposed HBA achieves a better throughput by averaging the values (values improved by 32.83%) when compared to LEACH.

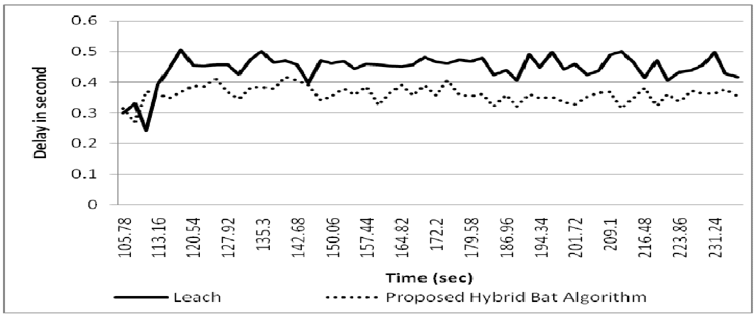

Figure 3: Delay (s) vs Time (Sec)

Figure 3 depicts that the proposed HBA lowers the delay better by averaging (lower by 18.91%) when compared to LEACH.

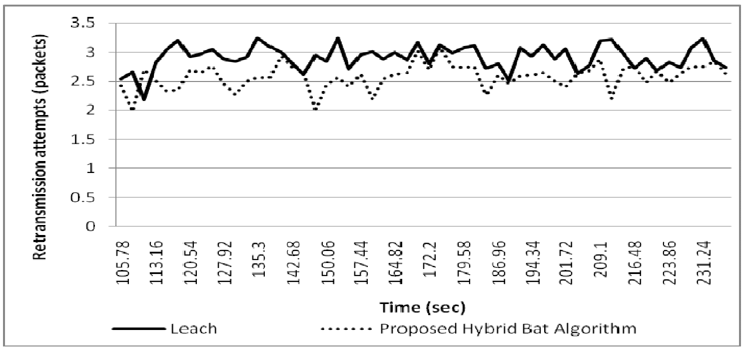

Figure 4: Retransmission Attempts (Packets) vs Time(Sec)

Figure 4 shows that the proposed HBA more efficiently lowers the retransmission of the packet - by averaging (lowers by 11.46%) when compared to LEACH.

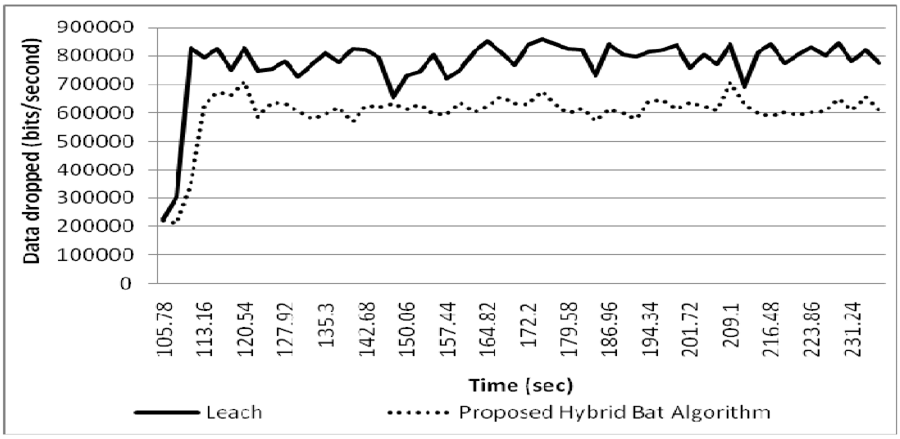

Figure 5: Data Dropped (Bits/Second) vs Time(Sec)

Figure 5 shows that the proposed HBA lowers the data dropped of the packet more efficiently - by - averaging (lowers by 22.2%) when compared to LEACH.

CONCLUSIONS

In this study, an HBA with (DE) to improve the efficiency and to overcome the disadvantages of LEACH was proposed. LEACH was enhanced by the use of the hybrid BAT method for cluster formation and CH selection based on the

distances and the number of CHs. Simulation revealed that the new HBA achieved better throughput and lowered the delay, data dropped, and packet retransmission. The new HBA achieved improved throughput (by 32.83%), better lowering of the delay (by 18.91%), better lowering of packet retransmission (by 11.46%)and better lowering of the data packets' dropping (by 22.2%) by averaging the values in comparison with the LEACH.

REFERENCES

1. Buratti, C., Conti, A., Dardari, D., and Verdone R., An overview on wireless sensor networks technology and evolution. *Sensors*, 2009, (9), 6869-6896.
2. Lee, W. L., Datta, A., and Cardell-Oliver, R. Network management in wireless sensor networks. *Handbook of Mobile Ad Hoc and Pervasive Communications: American Scientific Publishers*.2006
3. Akkaya, K., and Younis, M. A survey on routing protocols for wireless sensor networks. *Ad Hoc Networks*, 2005, *3*(3), 325-349.
4. Azahari, A., Alsaqour, R., Uddin, M., and Al-Hubaishi, M. Review of error detection of data link layer in computer network. *Journal of Engineering and Applied Sciences*, 2014, 20 *9* (1).
5. Diwan, R., and Thakur, S. Role of data link layer in OSI. *International Research Journal*, *1*(7), ISSN-0975-3486, 2009, RNI: RAJBIL 30097.
6. Lee, G., Kong, J., Lee, M., and Byeon, O. A cluster-based energy-efficient routing protocol without location information for sensor networks. *Journal of Information Processing Systems*, *1*(1), 49-54.
7. Shukla, K. V. Research on energy efficient routing protocol LEACH for wireless sensor networks. *International Journal of Engineering Research and Technology*, (*2*(3)) ESRSA Publications.
8. Jiang, C., Xiang, M., and Shi, W.Overview of cluster-based routing protocols in wireless sensor networks. *Electric Information and Control Engineering (ICEICE), International Conference, 2011* pp. 3414-3417. IEEE.
9. Reddy, N. G., Chitare, N., and Sampalli, S. Deployment of multiple base-stations in clustering protocols of wireless sensor networks (WSNs). *Advances in Computing, Communications and Informatics (ICACCI), International Conference*, pp. 1003-1006. IEEE. (2013)
10. Sharma, T., Kumar, B., Berry, K., Dhawan, A., Rathore, R. S., and Gupta, V. Ant based cluster head election algorithm in wireless sensor network to avoid redundancy. *Communication Systems and Network Technologies, 2014 Fourth International Conference*, pp. 83-88. IEEE. (2014)
11. Kumar, G., and Singh, J. Energy efficient clustering scheme based on grid optimization using genetic algorithm for wireless sensor networks. *Computing, Communications and Networking Technologies, Fourth International Conference*, pp.1, 4-6. IEEE. (2013)
12. Li, Q., Cui, L., Zhang, B., and Fan, Z. A low energy intelligent clustering protocol for wireless sensor networks. *Industrial Technology, 2010 IEEE International Conference*, pp. 1675-1682, IEEE. (2010)

13. Rejina P. J., and Vasanthanayaki, C. Gravitational search algorithm based mobile aggregator sink nodes for energy efficient wireless sensor networks. *Circuits, Power and Computing Technologies, 2013 International Conference, 2013,* pp. 1052-1058 IEEE.

14. Siew, Z. W., Chin, Y. K., Kiring, A., Yoong, H. P., and Teo, K. T. K. Comparative study of various cluster formation algorithms in wireless sensor networks. *Computing and Convergence Technology, 2012 Seventh International Conference, 2012*, pp.772-777, IEEE.

15. Fu, C., Jiang, Z., and Wei, W. W. A. An energy balanced algorithm of LEACH protocol in WSN. *International Journal of Computer Science*. (2013)

16. Li, Y., Yu, N., Zhang, W., Zhao, W., You, X., and Daneshmand, M. Enhancing the performance of LEACH protocol in wireless sensor networks. *Computer Communications Workshops* (INFOCOM WKSHPS), 2011 IEEE Conference, 2011, pp. 223-228, IEEE.

17. Goyal, S., and Patterh, M. S. Wireless sensor network localization based on BAT Algorithm. *International Association of Scientific Innovation and Research*, 2013, ISSN, 2279-0055.

18. Yilmaz, S., and Kucuksille, E. U. Improved bat algorithm (IBA) on continuous optimization problems. *Lecture Notes on Software Engineering*, 2013, *1* (3), 279-283.

19. Yang, X. S., and He, X. Bat algorithm: literature review and applications. *International*

20. *Journal of Bio-Inspired Computation*, 2013, *5* (3), 141-149.**)**

21. Ardia, D., Boudt, K., Carl, P., Mullen, K. M., and Peterson, B. G., Differential evolution with de optim. *R Journal*, 2011, *3* (1), 27-34.

22. Qin, A. K., Huang, V. L., and Suganthan, P. N., Differential evolution algorithm with strategy adaptation for global numerical optimization. *Evolutionary Computation*, IEEE Transactions on, 2009, *13* (2), 398-417.

23. Fister Jr., I., Fister, D., and ang, X. S. A hybrid bat algorithm. arXiv preprint arXiv:1303.6310. 2013